Undefined 1 (2015) 1–5
IOS Press

A Pattern Recognition Approach for Peak Prediction of Electrical Consumption [1]

Morten Goodwin [a,*], Anis Yazidi [b]

[a] *Department of ICT, University of Agder, Grimstad, Norway and Teknova AS, Grimstad, Norway*
E-Mail: morten.goodwin@uia.no
[b] *Institute for ICT, Oslo and Akershus University College of Applied Sciences, Oslo, Norway*
E-mail: Anis.Yazidi@hioa.no

Abstract.
Demand peaks in electrical power consumptions pose serious challenges for energy companies as these are typically unforeseen and require the net to support abnormally high consumption levels. Such peaks can be regulated in smart energy grids with the introduction of relatively simple techniques such as load balancing and smart pricing strategies. This is, however, difficult in practice because it requires prediction of peaks prior to their actual occurrence.

While most studies formulate the problem as an estimation problem, we take a radically different approach and formulate it as a classical pattern recognition problem. Further, the paper applies classification methods to solve the problem and applies these with real-life data from a Norwegian smart grid pilot project. Some of the key findings are that the algorithms can accurately detect 80% of energy consumption peaks up to one week ahead of time. Furthermore, we introduce a novel Learning Automata based approaches for selecting the optimal prediction model from a pool of models in an online fashion.

Keywords: Peak prediction, Power Consumption, Classification, Pattern Recognition

1. Introduction

Electricity consumption peaks appear seemingly random in the consumption data as a consequence of collective behaviour of several customers, and which in turn is influenced by many external factors. A simple example is the aggregate behavior of many consumers turning on their home heaters simultaneously, within a short time span, as a consequence of a cold weather wave. This aggregate behavior is natural since a drop in temperature affects a large population which might cause a peak in the overall electrical consumption. However, there are many factors other than merely temperature that are susceptible of influencing a user's electrical consumption: thus, it is not straightforward to foresee what the consumption level and in turn predict high loads.

Both consumption peaks and supply peaks cause serious challenges for electrical service providers because they need to over-dimension their grid in order to support the maximum potential load. Supporting these peaks is almost vital for energy companies since energy scarcity can leade to severe consequences such as power outages. An alternative approach to circumvent these peaks and reduce the over-dimensioning cost and magnitude is to balance them out with the introduction of intelligent methods for controlling them. Controlling the peaks can be done in several ways, such as performing load balancing and developing dynamic and intelligent pricing strategies. The later helps to reduce the peak magnitude and duration since consumers are

[1] A preliminary version of this paper that does not include the meta-learning algorithm (as well as the theory of learning automata) introduced in this journal paper was presented at AIAI'14, the 2014 IFIP WG 12.5 International Conference on Artificial Intelligence Applications and Innovations, Rhodes, Greece, in September 2014. We are very grateful to the anonymous Referees of the previous version of the paper, whose suggestions greatly improved the quality of this version.

[*] Corresponding author. E-mail: morten.goodwin@uia.no.

sensitive to changes in electrical prices and, generally, try to generally reduce the “unnecessary" part of their electrical consumption whenever the electricity price is high. In addition, smart grid systems allow appliances to access the electricity price and consequently can issue autonomous decision for turning on “non-critical load" as a consequence of price increase.

Despite the fact that the later intelligent control strategies are appealing, their success in reducing the magnitude and duration of the consumption peaks is entirely dependent on the ability to accurately predict their occurrence in advance, which is a particularly difficult as peaks are influenced by complex behaviour and typically occur in abnormal situations.

This paper addresses predicting electrical power consumption peaks in small communities by relying solely on historical consumption and without resorting to external data sources such as weather predictions.

The output peak prediction algorithms presented in this paper are intended to be used in a pilot smart grid installation in order to achieve the following automatic behavior at the customer side: (1) automatically turning off and on smart electrical appliances, (2) and carrying out local level load balancing.

This article presents two fold contribution. First, we map peak detection into a two-labelled classification problem. Second, we propose a meta-learning algorithm based on the theory of Learning Automata that combines many underlying classifiers for predicting the peak. The Learning Automata algorithm is a modification of the Fast Learning Automata algorithm reported in [28] and is shown to outperform it in the experimental results.

We, further, test our algorithm out with data from a real Norwegian pilot smart grid project, and are able to obtain results in line with state-of-the-art with relatively simple classification algorithms.

This paper is organised as follows. Section 2 formally maps the problem to be solved as a two-labelled classification problem. A meta-learning algorithm is then devised as to choose the optimal prediction model from a pool of models in an online manner. Section 3.2 gives first an overview of the data used in these experiments and then continues with the results and findings from these methods.

Section 4 states the most relevant development in the area from two research areas: peak prediction/detection and rare item classification.

Finally, conclusions and future work is outlined in section 5.

2. Proposed Solution

In this section, we shall present our solution to the peak detection problem which consists of two main components:

- Mapping the problem to two-label classification problem which is described in section 2.1. In order to achieve this goal, we will present different relatively simple classifiers that fit this purpose.
- A meta-learning algorithm based on theory of Learning Automata for selecting the best prediction classifier in an online manner. The meta-learning algorithm is presented in section 2.2.

2.1. Problem setting and modelling

The goal of this research is to predict load peaks prior to their occurrence. Formally, given a time sequence **t**, let $p(t_i, n)$ be a statistical function asserting whether t_i is a peak where $t_i \in \mathbf{t}$ and n tells the extremity of the peak as follows:

$$p(t_i, n) = \begin{cases} \textit{peak if } p(t_i, n) > \mu + n * \sigma \\ \textit{not peak otherwise} \end{cases} \quad (1)$$

where μ and σ are the arithmetic average and standard deviation of **t**, and n is a number defined as the extremity of the peak. Thus, a value is defined as a peak if it overpasses the the mean value with more than n times the standard deviation. The main question we deal with here is predicting the output from $p(t_i, n)$ using only part of the sequence occurring prior to t_i ($t_{<i} \in \mathbf{t}$). Hence, the peak detection problem is reduced to a standard two-labelled classification problem. This is less trivial than most two-labelled classification problems because the peaks are rare and “abnormal”.

In layman’s terms: Is it possible to predict future load peaks using only past data on electrical consumptions?[1]

This sections presents approaches for to predicting $p(t_i, n)$. The first approach is a straightforward solution introduced for comparison purposes. This is continued with standard classification techniques in-

[1]Note that it is very different from the seemingly similar peak detection (positive outlier detection). For outlier detection it is possible to calculate whether t_i is an outlier using t_i itself.[12]

spired by similar work on load forecasting [10,23,19, 6]. Lastly, we present hierarchical classification solutions inspired by techniques presented in the literature on rare item classification [17,35,45]. All classifiers aim at predicting $p'(t_{<i}, n)$ as close to $p(t_i, n)$ as possible.

Majority voting(c_m):Firstly, a simple majority voting classifier was implemented that counts the number of peaks in the training data and checks whether there are more peaks than the previous day than could be expected based on simple statistics was implemented.

Consistent peaks(c_c):Secondly, another classifier was implemented that always predict the peak equal to the previous peak value as following:

$$c_c(t_{<i}, d, n) = p'(t_{<i}, n) = p(t_{i-d}, n). \quad (2)$$

where t_i is the data to predict and d indicates how many hours in advance t_i is.

SVM(c_l,c_p):The purpose of the Support Vector Machines SVM is to create a function, linear(c_l) or polynomial(c_p), that separates data that results in peaks from the data that does not. Upon predicting $p(t_i, n)$ the vector space consists of a defined number of points prior to $t_i \in \mathbf{t}$, namely $t_{<i}$. Each individual point in $t_{<i}$ is then a separate dimension. Two variants of SVM are implemented; the linear SVM approach is defined as $c_l(t_{<i}, d, n)$, while the polynomial is defined as $c_p(t_{<i}, d, n)$.

GMM(c_g):The The Gaussian Mixture Model Classifier (GMM) is a classifier known for handling outliers well [16]. It has, to the best of our knowledge, not been used for load or peak prediction previously. It has however been used for many other classification applications with success, and is specially suited for rare item classification. Similarly to the SVM classifiers, when we perform classification of the load at t_i with GMM, we populate the vector space with points prior to t_i ($t_{<i}$). This classifier is defined as $c_g(t_{<i}, n)$.

Two variants of hierarchical classifiers are presented: Hierarchical and Hierarchical with Consistent Peaks. They both use a flat approach similar to variants in the literature [45] — namely straightforward combinations of the other classifiers.

Hierarchical (c_{h1}): The hierarchical classifier is simply a combination of the linear SVM and GMM similar to the common practice in the literature as follows:[41,14].[2]

$$c_{h1}(t_{<i}, d, n) = c_l(t_{<i}d, , n) \mid c_g(t_{<i}d, n) \quad (3)$$

Hierarchical with Consistent Peaks (c_{h2}):An additional classifier was created utilizing the data from the consistent peaks (see equation 2). This extends equation 3 as follows:

$$c_{h2}(t_{<i}, d, n) = c_l(t_{<i}, d, n) \mid c_g(t_{<i}, d, n) \mid c_c(t_{<i}, d, n) \quad (4)$$

2.2. Meta-Learning algorithm based on Learning Automata

The fundamental tool which we shall use in this section involves Learning Automata. Learning Automata (LA) have been used in systems that have incomplete knowledge of the environment in which they operate [1,18,25,26,27,32,39]. The learning mechanism attempts to learn from a *stochastic Teacher* which models the Environment. In his pioneering work, Tsetlin [40] attempted to use LA to model biological learning. In general, a random action is selected based on a probability vector, and these action probabilities are updated based on the observation of the Environment's response, after which the procedure is repeated.

The term "Learning Automata" was first publicized and rendered popular in the survey paper by Narendra and Thathachar. The goal of LA is to "determine the optimal action out of a set of allowable actions" [1]. The distinguishing characteristic of automata-based learning is that the search for the optimizing parameter vector is conducted in the space of probability distributions defined over the parameter space, rather than in the parameter space itself [38].

In this section, we adopt the Fast Learning Automata (FLA) algorithm proposed in [28] because of its low complexity. In addition, we propose a variant of the FLA which we call "unbalanced FLA" and show its superior performance when compared to the FLA in our experimental settings.

2.2.1. Balanced FLA Algorithm Applied to Peak Prediction

We shall formally define the FLA algorithm and explain how it is linked to our problem [28]. In the rest of this paper, we will refer to the original FLA

[2]Note that the decision rule is simpler than most other setups.

as "balanced FLA" due to Obaidat et al. [28]. Let $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ be the set of outputs or actions that the LA must choose from, and $\alpha(t_i)$ is the action chosen by the automaton at any instant t_i.

We suppose that each action α_k corresponds to a prediction model in our settings.

We denote by $S = (a_1(t_i), a_2(t_i), \ldots, a_r(t_i))$ the set of non-normalized probability vector [28]. For $k \in \{1, 2, \ldots, r\}$, and $\forall\, t_i$, we suppose that $a_k(t_i)$ takes N integer values from the set $\in \{1, 2, \ldots, N\}$. The later condition is guaranteed owing to the update form of the FLA which we shall describe later in this section.

P denotes the normalized probability vector $P = (p_1(t_i), p_2(t_i), \ldots, p_r(t_i))$ constructed using S in the following manner: at time instant t_i, an action $k \in \{1, 2, \ldots, r\}$, is chosen according to the normalized probability: $p_k(t_i) = \frac{a_k(t_i)}{\sum_{j=1}^{r} a_j(t_i)}$.

When an action k is chosen, the Environment gives out a response $\beta(t_i)$ at a time "t_i". The automaton is either penalized or rewarded. On the basis of the response $\beta_k(t_i)$, we update a_k for the next time instant according to:

- $a_k(t_{i+1}) = a_k(t_{i+1}) + 1$ if $\beta_k(t_i) = 1$ and $a_k(t_{i+1}) < N$.
- $a_k(t_{i+1}) = a_k(t_{i+1}) - 1$ if $\beta_k(t_i) = 0$ and $a_k(t_{i+1}) > 1$.

It is worth emphasizing that, based on the above update form, the smallest possible value for $a_k(t_i)$ is 1, which is different from the classical FLA proposed by Obaidat and his colleagues [28], since in the later work the lowest value corresponds to 0 . Therefore the current FLA, is ergodic, which allows the scheme to cope with dynamic environments where the optimal action changes over time. Please note that the original FLA falls into the family of absorbing Learning Automata.

Note that the highest value for $p_k(.)$ is achieved whenever $a_j = 1$ for $j \neq k$ while $a_k = N$, which corresponds to $p_k(.)$ equal to $\frac{N}{(r-1)+N}$.

Let $e_k(t_i)$ denote the prediction error of model k at time t_i as following:

$$e_k(t_i) = |p'(t_{<i}, n) - p(t_i, n)| \tag{5}$$

where $p'(t_{<i}, n)$ depends on the model k used. $p'(t_{<i}, n)$ is the predicted value at the next time instant t_i, $p(t_i, n)$ is the ground truth that we can only observe at the next time instant, and n is the extremity of the peak (see equation 1).

Now let us define the feedback $\beta_k(t_i)$ for a chosen action k.

$\beta_k(t_i) = 1$, i.e, reward, if the chosen action k, corresponding to the prediction model k, achieves the lower prediction error than the predictions models associated to the other $r - 1$ actions α_j where $j \neq k$. Similarly, $\beta_k(t_i) = 0$ i.e, penalty, if the chosen action k does not correspond to the lowest prediction error.

In formal terms:

- $\beta_k(t_i) = 1$ if $e_k(t_i) = min(e_1(t_i), e_2(t_i) \ldots e_r(t_i))$
- $\beta_k(t_i) = 0$ if $e_k(t_i) \neq min(e_1(t_i), e_2(t_i) \ldots e_r(t_i))$

Where min(.) denotes the minimum.

2.2.2. Unbalanced FLA Applied to Peak Prediction

We propose the so-called "unbalanced FLA" where we give different weights to reward an penalty.

When an action k is chosen, the environment gives out a response $\beta(t_i)$ at a time "t_i". The automaton is either penalized or rewarded in an "unbalanced manner". On the basis of the response $\beta_k(t_i)$, we update a_k for the next time instant according to:

- $a_k(t_{i+1}) = max(a_k(t_{i+1}) + R, N)$ if $\beta_k(t_i) = 1$.
- $a_k(t_{i+1}) = a_k(t_{i+1}) - 1$ if $\beta_k(t_i) = 0$ and $a_k(t_{i+1}) > 1$.

Philosophically, the FLA is akin to the "Reward Penalty" LA since Reward and Penalty are given the same update weight, i.e, incrementing or decrementing the non-normalized probability with 1 state. In fact, the FLA increases or decreases the state by 1 upon receiving a reward or a penalty. On the other hand, the "unbalanced FLA" is akin to "Reward ϵ Penalty" Learning Automata, as we give more weight to the reward by incrementing the non-normalized probability component at most by R states instead of 1 state.

3. Experimental Results

3.1. Data with electrical consumption peaks

The data used in this project was collected during six weeks in mid 2012 from a summer cabin area in Hvaler, Norway [34]. In total, there are more than 7900 installations with varying degree of activity — introducing fluctuations into the data collected. From each installation, accumulated hourly energy consumptions (kWh) was collected throughout the six weeks. The network operator extracted and quality-assured the metering data, and made it available in GS2 ver. 1.2 format — a standard format for exchange of metering data in Norway and Sweden [33]. A script was cre-

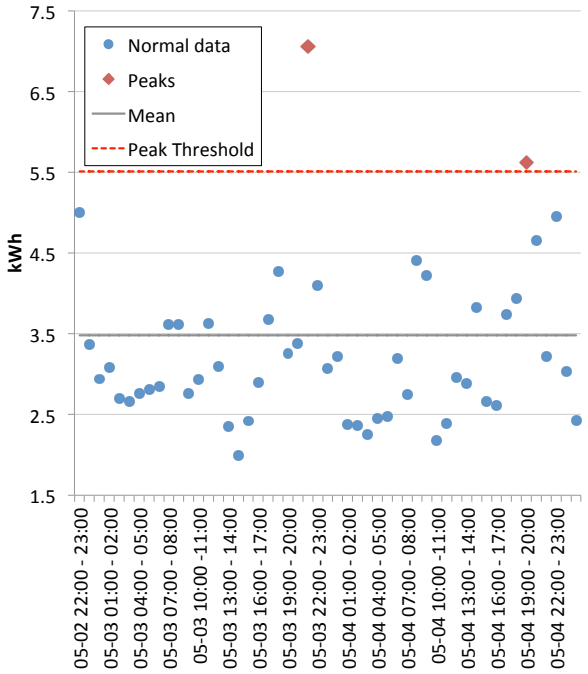

Fig. 1. Example of peak in the consumption data with n set to 2

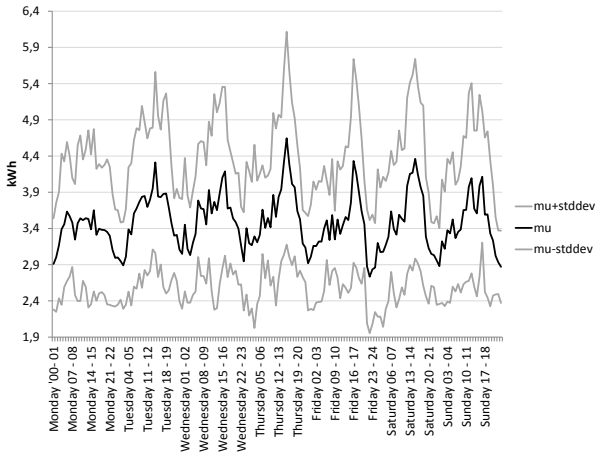

Fig. 2. Consumption averaged per 7 days

ated to reformat the data from GS2 to Python structure to make it easy to train and verify the classification schemes.

An example of electrical consumption for one installation is presented in Figure 3.1. This shows averaged values of 7 days — displaying some clear trends: a rise and fall of the consumption with its highest magnitude takes place around 17:00 at Norwegian dinner time and with it lowest magnitude at night. The pattern is repeated with a daily frequency.

Figure 1 shows an example of consumption with peaks for a specific 24 hour period. The threshold level, n, for the peak data in Figure 1 is set to 2 yielding two peaks: around 05-03 22:00 and 05-04 16:00. Figure 3.1 consumption averaged over 7 days showing an interesting cyclic behvaoir in the data. Overall, by varying n in the equation 1, the percentage of peaks varies from 15% for $n = 1$ down to 0.8% for $n = 3$.

3.2. Experiments

This section presents empirical results from running the predictive algorithms introduced in section 2. In line with common practice in the field [47], we present predictions starting from one hour into the future and up to 7 days (168 hours).

Section 3.2.1 present comparative experiments where n set to 1 — predicting moderate peaks. Additionally, section 3.2.2 shows the behavior of the algorithms when n increases — making the peaks to detect more extreme.

All results presented in this chapter use a vector size of 24.[3]

Further, several metrics exist to evaluate information retrieval approaches. This paper promotes classification methods that favour false positives over false negatives. This is because a false positive, predicting a peak that is not present, has significantly fewer consequences than not predicting peaks which are present. Thus, high recall is much more favored than high precision.

3.2.1. Predicting small consumption peaks

This section presents peak predictions with n in equation in $p(t_i, n)$ (equation 1) set to 1. This is the most straightforward approach where peaks are defined as simple values larger than the standard deviation.

Figure 3 and 4 show the recall and precision of the applied algorithms. All these figures show predictions 1 to 168 (7 days) into the future with n set to 1. This means that at each time instance, the algorithms try to predict from 1 to 168 hours ahead of time and the average is shown in the graph.

We observe that all pattern recognition algorithms, SVM, GMM and Hierarchical have a decrease in recall

[3]Several experiments were carried out with various vector sizes concluding with a vector size of 24 units. These results are not crucial to the understanding of the approach and, therefore, omitted.

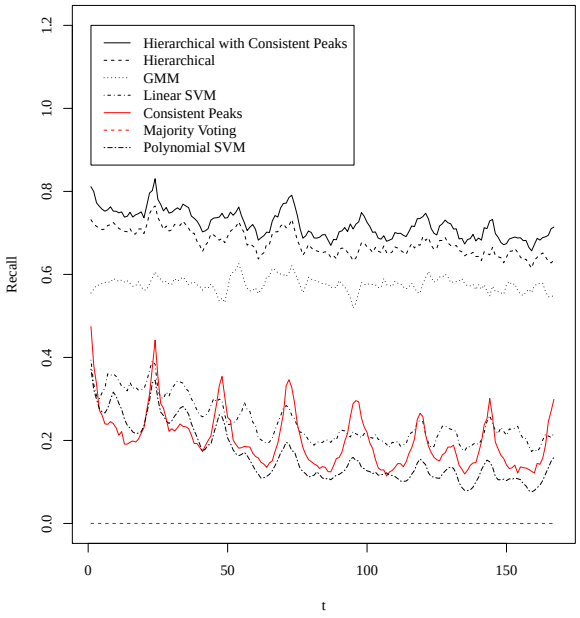

Fig. 3. Recall for predicting 1 to 168 hours into the future with n set to 1

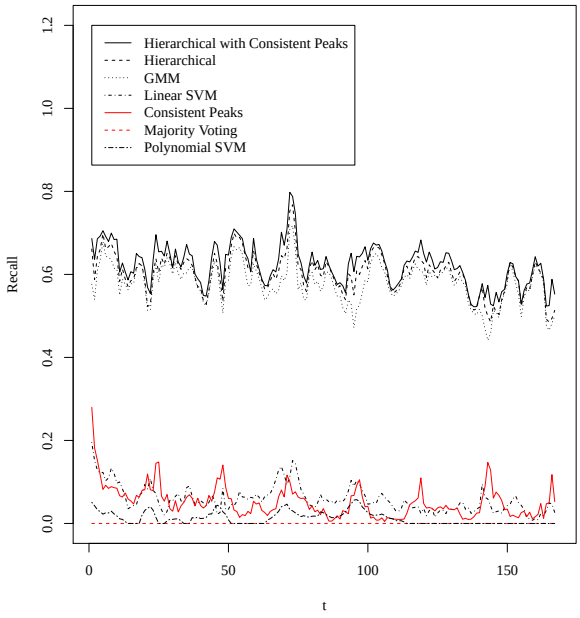

Fig. 5. Recall predicting 1 to 168 hours into the future, n set to 2 yielding 35 732 peaks.

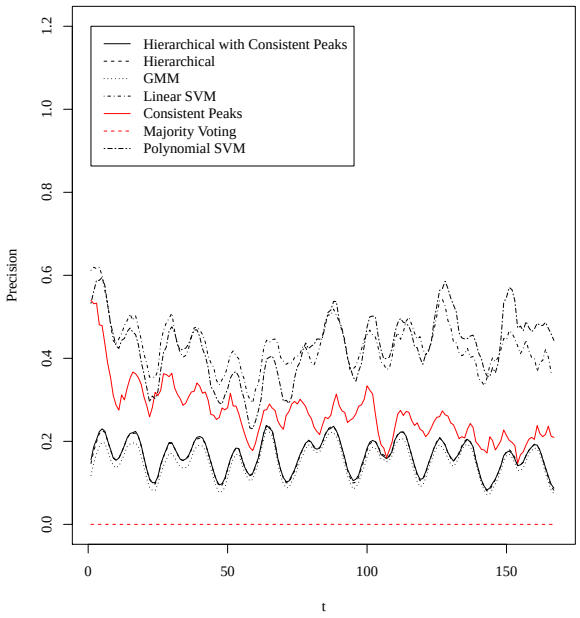

Fig. 4. Precision for predicting 1 to 168 hours into the future with n set to 1

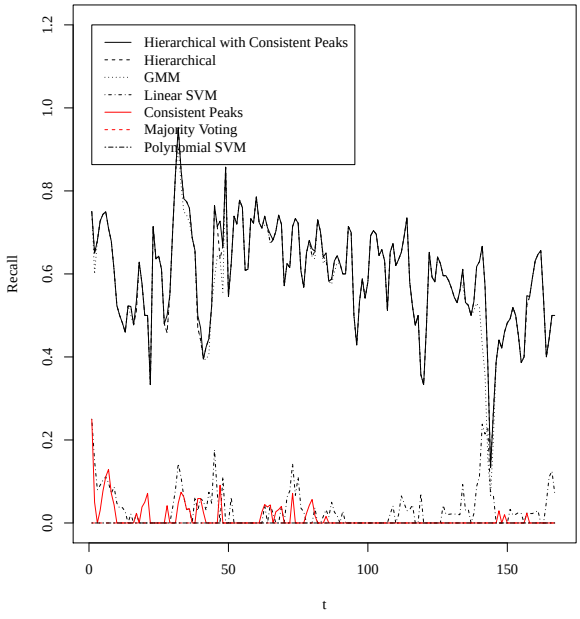

Fig. 6. Recall predicting 1 to 168 hours into the future, n set to 3 yielding 7 614 peaks.

as the t increases. This means that, intuitively, the difficulty increases whenever the algorithm tries to predict further into the future. However, when t reaches 24, the quality of the measurement increases. Hence, it is easier to predict 24 hours into the future than it is to predict other values. This suggests that people follow roughly the same patterns in 24 hour cyclic intervals (cook dinner, eat, sleep at roughly the same intervals). This is supported by "Consistent peaks"-classifier which is surprisingly accurate.

The figure shows that the hierarchical classifiers have an excellent recall (Figure 3). Thus, these algorithms are able to detect about 80% of the peaks in the data. Further, the recall decreases as t increases for all algorithms. Hence, prediction difficulty arises whenever the algorithms look further into the future. This is, however, less true for hierarchical classifiers and GMM than for comparative algorithms.

On the other hand, the hierarchical classifiers have a low precision (Figure 4). The algorithms have many false positives (close to 80%). We can say that an hierarchical classifier is able to predict 80% of the peaks. Moreover, whenever it predicts a peak there is a 20% chance that it is an actual peak. It is crucial to emphasize that the objective of the algorithms is to favor false positives over false negatives false negatives, which have less impact in electrical power grids.

3.2.2. Predicting larger consumption peaks

This section presents experimental results in which we vary n in $p(t_i, n)$ (equation 1). By increasing n, the peaks become more extreme and (presumably) more difficult to predict.

Figures 5 and 6 depict recall over time when n is fixed to 2 and 3 respectively. This is the same setup as shown in figure 3 with n set to 1, but the behaviour is notably different. First and foremost, all classifiers have a decreased recall. Most notably, the SVM classifiers drop fast to a recall close to 0. In this scenario, SVM is only able to predict close to 24 hours into the future with a poor performance. Consequently, the hierarchical classifiers are only slightly better than GMM — which means that SVM does not contribute significantly to an increased recall in the hierarchical classifiers.

The trend in Figure 3 (n=1) and Figure 5 (n=2) is continued in Figure 6 (n=3), but is more extreme. The SVM classifier drops more quickly to 0, and the difference between the hierarchical and GMM classifiers is considerably smaller. Nonetheless, the best algorithms are able to reach a recall of more than 70%. [4]

Another trend which is visible as n increases is that the results become more chaotic (from Figure 3 (n=1), to Figure 5 (n=2), and Figure 6 (n=3)). A possible reason for this could be that when n increases and the peaks become more extreme, the peaks become more difficult to predict — and in turn the results appear more chaotic.

A conclusion worth drawing is that an increase in the extremities of the peaks makes the classification more difficult. However, GMM and corresponding hierarchical classifiers are still able to detect more than 70% of the peaks while all the other classifiers prove insufficient with a recall close to 0.

Table 1 show the corresponding ROC curve. It shows that when the SVM classifiers and consistent peaks have high false positive rate, with a high n, they have a low true positive rate. The hierarchical classifiers are more linear.[5]

3.3. Experiments Under Different Prediction Models: Meta-Learning Algorithm

In the previous section, we have shown that it is possible to predict future load peaks using only past data on electrical consumptions.

However, the question that we are trying to answer here by resorting to LA is: how to choose the "ideal" prediction model from a set of regression models in an online and adaptive manner. We achieve this by creating a "competition" between the different models using the theory of LA. The rationale for resorting to LA is that: the model that achieves the highest reward will be chosen more often, thus, the prediction error of the LA will decrease and will approach the prediction of the most accurate model.

This idea is similar to the well-known research research on Weighted Majority Voting. However, the LA prediction is not based on weighted combination of the available prediction models, but on the prediction model that is selected by the LA at each time instant. The LA attempts to learn over time the prediction model that achieves the highest reward. The models included in the study are SVM (Linear Support

[4] The trend continues when increasing n even further. However, this has been deliberately excluded from of the paper since this does not contribute to further insight into algorithms.

[5] A ROC curve is often plotted. However, due to the low data it communicated better in a table.

Peak Size	GMM		Hierarchical		Hirarchical with Consistent Peaks		Linear SVM		Polynomial SVM		Consistent Peaks	
	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR
1	0.63	0.60	0.62	0.60	0.63	0.61	0.98	0.91	0.99	0.91	0.99	0.91
2	0.30	0.28	0.30	0.28	0.30	0.28	0.94	0.67	0.97	0.69	0.96	0.69
3	0.09	0.08	0.08	0.08	0.08	0.08	0.76	0.34	0.80	0.34	0.79	0.35

Table 1

ROC for algorithms varying n with False Positive Rates (FPR) and True Positive Rates (TPR)

Vector Machine with kernel parameter 0.1, regularization parameter 1.0, tolerance for testing KKT conditions 0.001, and convergence parameter 0.001), Lars (Least Angle Regression), Ridge (Ridge Regression), and OLS (Ordinary Linear Least Square Regression with regularization parameter 1.0). The LA aims at selecting the most suitable among those four models. [6]

All results presented in this section use a vector size of 24. The aim is to to predict the consumption. We report the error for different values of prediction horizon. The error is computed for all the data set and is calculated as a time average. In addition, we use $N = 100$ as the number of states per action.

In Figure 7, we report the average prediction error for the "balanced FLA" for a time horizon varying from 1 hour to 24 hour ahead of time. We observe that for all the prediction models, including the "balanced FLA", the error increases as the time horizon increases. The explanation of the results is intuitive. In fact, the more ahead of in time a model aspires to predict the electricity consumption, the more error we observe. In addition, we observe that the "balanced FLA" performance approaches the SVM performance, where the latter is the best prediction model in this experiment. The reason why the"balanced FLA" model does not yield a performance identical to the SVM is that the fact that it was made "artificially" ergodic, which means that it does not get locked into choosing one action after some iterations.

In Figure 8, we observe the prediction error for different values of time horizon varying from 1 hour to 7 days. Again, we observe that the performance of all the approaches degrade as the prediction horizon increases.

Furthermore, we present the results of the "unbalanced FLA" with a parameter $R = 10$. We observe that the "unbalanced FLA" depicted in Figure 9 and Figure 10 achieves a higher performance than its counterpart "balanced FLA" depicted in Figure 7 and Figure 8.

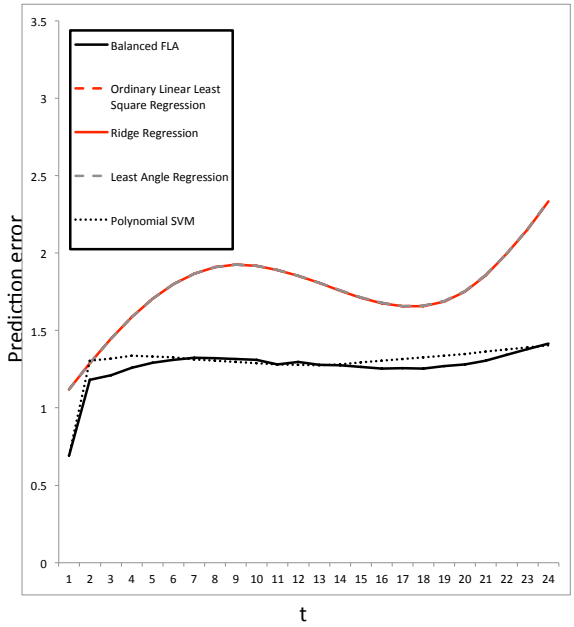

Fig. 7. Performance of "balanced FLA" based algorithm for predicting consumption is based on regression models which predict consumption 1 to 24 hours into the future.

3.4. *Experiments Under Different Parameterized SVM Models*

In this experiment, instead of choosing different prediction algorithms as actions for the LA, we choose different SVM prediction models parameterized with different vector sizes as our prediction models.

It is worth noting that the expriments from section 3.2 show that the SVM prediction exhibits good performance for a vector size of 24 hours (equivalently history of 24 hours).

In Figure 11, the same observation is made again through the use of LA. In fact, in Figure 11, we com-

[6] Note that the model can easily be extended with including different models. The purpose here is to show that the LA is adaptively use the "best" among the available models.

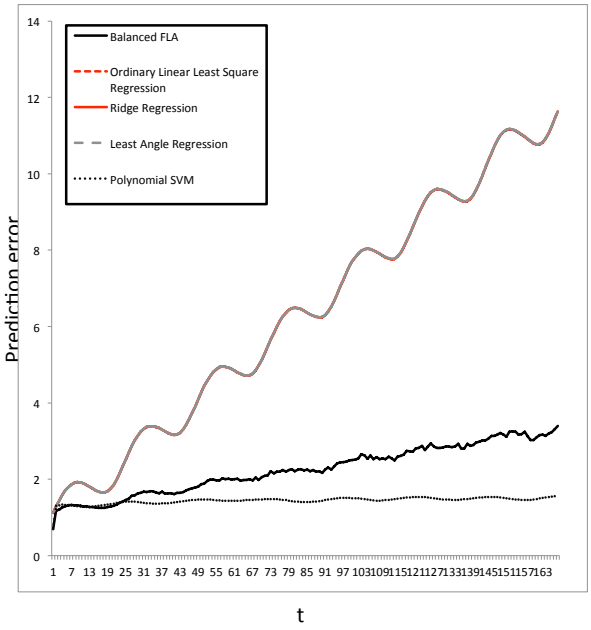

Fig. 8. Performance of "balanced FLA" based algorithm for predicting consumption based on regression models predicting consumption 1 to 168 hours (7 days) into the future.

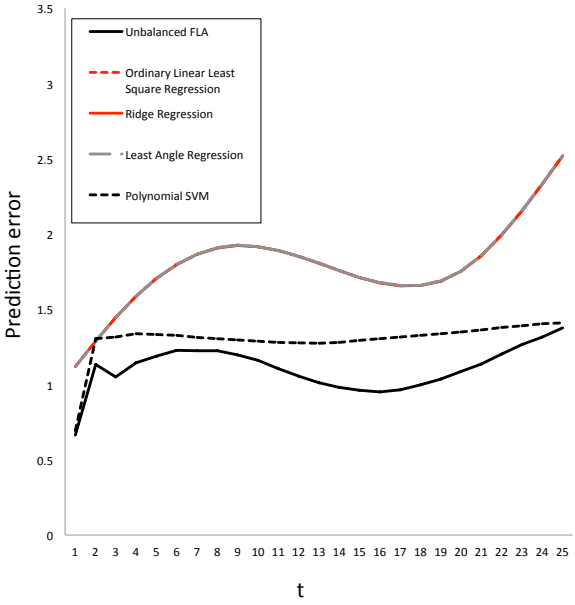

Fig. 9. Performance of "unbalanced FLA" based algorithm predicting consumption is based on regression models which predict consumption 1 to 24 hours into the future.

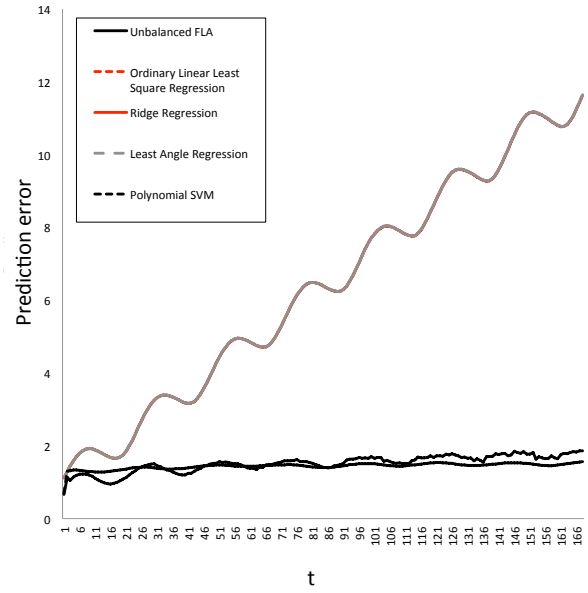

Fig. 10. Performance of "unbalanced FLA" based algorithm predicting consumption is based on regression models which predict consumption 1 to 168 hours (7 days) into the future.

pare our LA equipped with the following 4 actions: SVM model with vector size 6, 12, 18, and 24, approaches over time the performance of the best prediction model which is the SVM with a vector size of 24 hours. Again, since the LA is ergodic, it will not converge to get locked to choosing the best prediction model which achieves the lowest error, namely, the SVM model with vector size 24.

Furthermore, in Figure 12, we present the results for the "unbalanced FLA" with a parameter $R = 10$ for a time horizon of 24 hours. We observe that the "unbalanced FLA" outperforms all the the 4 predictions models: SVM model with vector size 6, 12, 18, 24.

4. Related Work

This section presents the most relevant research in the area of peak prediction. We have organized the related work into different categories. Section 4.1 presents the research on electrical loads prediction, and section 4.2 presents research on rare item classifications when there is a skewed division of labelled data — i.e. a classification when there are so few positive labels that leads to a significant negative impact on the classification results.

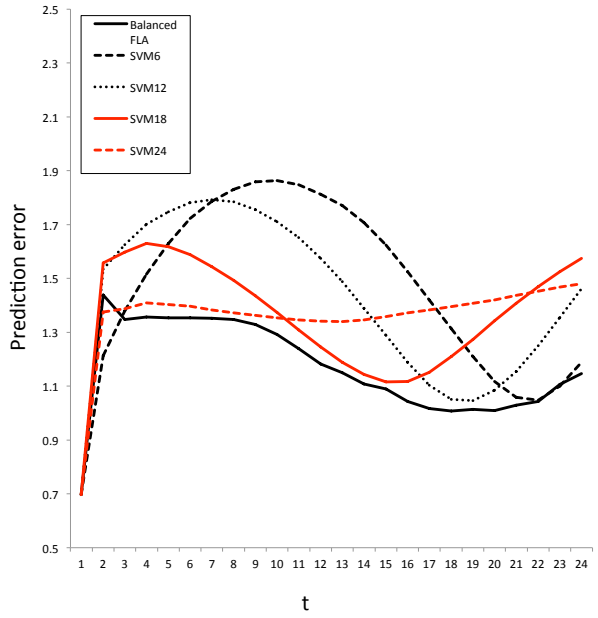

Fig. 11. Performance of the "balanced FLA" based algorithm predicting consumption is based on training history which predicts 1 to 24 hours into the future.

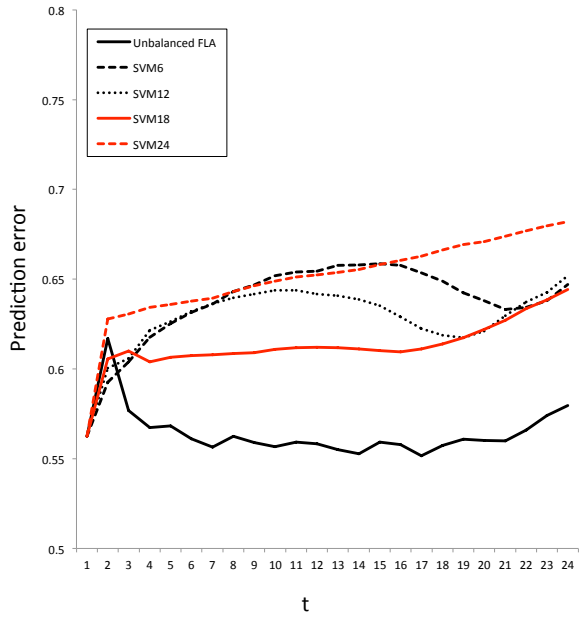

Fig. 12. Performance of the "unbalanced FLA" based algorithm predicting consumption is based on training history predicting 1 to 24 hours into the future.

4.1. Electrical consumption prediction

A significant amount of research has addressed load forecasting [9,4,44,24]. These studies have two common characteristics. First, they rely on certain third-party data for predictions. Second, they try to predict the actual power load. Despite the fact that peak prediction and load forecasting bear much similarity, their respective objectives are notably different.

In general, the techniques introduced in 4.1.1 and 4.1.2 use some third party data so as that to correlate it to the load and in turn predict the future consumptions, while the techniques shown in section 4.1.4 and 4.1.3 focus on predicting the actual load without identifying the peaks.

4.1.1. Regression models

The most common method used for peak prediction is regression. Regression aims to establish a statistical model for the load consumption that integrates many factors, such as customer profile, weather data etc. Based on the correlation between the energy load and the third party data, regression models are used to predict future peaks.

The most common and simple model for regression is the linear model, which is usually used in combination with other factors such as heat data. Linear regression model proves to be very useful for predicting short-term peak loads [10,29] and annual loads[43]. The later research uses support vector regression and various functional linear regression models for the actual prediction. It then clusters the data and assigns one regression model per cluster. The problem addressed in these studies is similar to the problem we addressed here. However, the the major difference between them is the reliance of regression models method on heat maps and the introduction of clustering phase.

Research on forecasting annual loads in combination with differential evaluation and support vector regression was reported in [43]. The authors of [43] resorted to differential evaluations so as that to choose parameters from an available predefined set of parameters while the regression is used for the actual forecasting.

4.1.2. Daily correlation based methods

A very common and intuitive approach of to load forecasting is to merely find a similar day based on the available data, such as weather reports, and assume that similar days have similar loads. A representative study that falls into this category is [15]. The later study uses a hierarchical two-step classifiers for short-

term load forecasting which first predicts the "type of day" then performs an hourly forecast from this output.

4.1.3. Internal structure

There is a vast amount of work that does not rely on a third party data but rather works under the premise that the consumption data exhibits some internal structure that can be learnt.[7] Some of the important studies that fall into this category include [47] which combines evolutionary algorithms and fuzzy logic for predicting hourly peaks from one to seven days ahead in time[47], wavelet transformation techniques to accurately forecast power consumptions about 4 hours ahead of time [2] and short-term predictions using exponential smoothing methods[37]. Further, other machine learning techniques have been applied such as non-linear curve-fitting [48,36] and Support Vector Machines (SVM) [23,19]. It should be noted here that the load forecasting is significantly different from peak prediction [30] [37].

4.1.4. Other Machine learning techniques

Additionally, several other machine learning techniques has been used for load forecasting.

Using neural networks which combine weather data and historical loads with non-linear curve-fitting has been popular [30,48,36]. Curve fitting was also used with fuzzy logic techniques [22,4] which avoids advanced mathematical models. The support vector machines (SVM) method is also a common methods for load forecasting [23,19], including daily load demands [6]. For this, the data is mapped in a multi-dimensional space and the problem is reduced to defining a kernel that discriminated the data classes from each other.

Other machine learning techniques that are common for load prediction include Kalman filters [20], recurrent neural networks [49], hidden Markov models [11], parallel classifiers [5] and firefly colony algorithm [8].

The advantage of using machine learning for load prediction in general is its simplicity as it does not requires advanced mathematical modelling. In addition, it can give relatively precise outputs.

It is worth mentioning that machine learning has been applied to solve other problems within the field of smart grid such as biding in electricity markets [31] and energy efficient routing [21].

[7] This is the same assumption we present in this paper.

4.2. Rare item classification

Rare item classification is a field within pattern recognition where the majority of the data is impertinent and consequently the interesting data appears much more rarely. In many situations, even the labels to be classified are not known in advance. Further, there is an imbalance in the data available for training (a priori). Exhaustive labelling is commonly required to develop up a proper training database. Both labelling and corresponding classification is inevitably costly. There is a significant amount of research that covers this topic; however, to the best of our knowledge, none of these directly address the peak prediction problem.

An effective technique when dealing with rare item classification is to combine different methods, such as hierarchical classifiers with more than one classification technique [17,35]. Generally, this reduces the problem into to organising local classifiers in an hierarchical structure and defining rules for issuing a global consensus. Based on techniques such as majority voting, the hierarchical classifier yields significantly better results for rare item classification than comparable algorithms, even with a flat structure [45]. Other methods use clever re-sampling, such as random under- and over-sampling [17] or stratified sampling [42].

There are many variants of rare item classification methods [35] which include, among others, hierarchical structures of SVM classifiers where each classifier has a different sampling scheme [46].

4.2.1. Classify with an unknown prior

Some of the existing work focuses on classification with non-parametric prior, i.e, the classes are not known [13,41,14].

The main methodology for solving this problem is composed of two main steps:

1. Discover and label classes based on unsupervised learning approaches such as clustering [13,41] or examining the variation of the data [14].
2. Train and classify using standard pattern recognition approaches, such as SVM and Gaussian Mixture Models (GMM).

Others techniques use an active learning approach where the supervision is minimized through a stream based active learning [7]. In this way, the authors are able to utilize reinforcement learning techniques for rare item classes avoiding separate training and classification phases. Similarly, in the active learning phases,

Dirichlet processes are used to calculate whether a sample is part of an unknown Dirichlet distribution.

4.3. Outlier detection

Much work has been carried out on outlier detection [12,3]. This research field resembles peak prediction in many ways. The most apparent similarity is that both problem focus on finding abnormalities in the data. However, the major difference between the two problems is that outlier detection has the complete data set available, which is not the case for peak prediction. SVM and GMM have successfully been applied to outlier detection [12], rare item classification (see section 4.2) and consumption prediction (see section 4.1).

5. Conclusion and future work

This paper addresses the issue of predicting electrical consumption peaks as input into load balancing and/or smart pricing strategies. This is done in a novel manner by mapping the prediction activities into a two-labelled classification problem. Further, in contrast to most existing approaches, the features are based solely on previous consumptions, avoiding reliance on third party data for the predictions.

Several classification algorithms are implemented and successfully applied to real-life data from a Norwegian smart-grid pilot project. The most promising results are produced by applying a simple hierarchical classifier with the combinations of linear support vector machines, Gaussian mixture models and deterministic assumption of consistent peaks. In fact, this solution is capable of to predicting 80% of the peaks, and whenever it predicts a peak it is a 20% chance that it is an actual peak. It is essential to remember that the objective of the algorithms is to favor false positives over false negatives, the latter, having less impact on electrical power grids. This is because a false positive, predicting a peak that is not present, has significantly fewer consequences than not predicting peaks which are present. Thus, high recall is much more favored after than high precision.

In addition, we present a meta-learning algorithm based on the theory of LA for selecting the optimal prediction model from a pool of models in an online fashion. Our LA algorithm is a modification of the fast LA algorithm reported in [28] where we give more weight to updates upon receiving reward and less weight to updates upon receiving penalty. This simple modification of the original balanced FLA is inspired by ideas for "Reward ϵ Penalty" LA which operates probability update under both reward and penalty response; however it gives more weight to the reward.

The properties of the LA approach is appealing since it does not require a priori knowledge of the best model, but rather explores the models space in search of the model that achieves the best performance.

6. Acknowledgements

This project is partially funded by Agder Regional Research Fund with the project “Grid Operation and Distributed Energy Storage: Potential for improved grid-operation efficiency” and the Norwegian Research Council funded project “Electricity Usage in Smart Village Skarpnes”. These projects are carried out together with University of Agder, Teknova, Agder Energi Nett and Eltek. Data is collected through the external DeVID project.[8]

References

[1] M. Agache and B. J. Oommen. Generalized pursuit learning schemes: New families of continuous and discretized learning automata. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 32(6):738–749, December 2002.

[2] Suseelatha Annamareddi, Sudheer Gopinathan, and Bharathi Dora. A simple hybrid model for short-term load forecasting. *Journal of Engineering*, 2013.

[3] Ira Assent, Philipp Kranen, Corinna Baldauf, and Thomas Seidl. Anyout: Anytime outlier detection on streaming data. In Sang-goo Lee, Zhiyong Peng, Xiaofang Zhou, Yang-Sae Moon, Rainer Unland, and Jaesoo Yoo, editors, *Database Systems for Advanced Applications*, volume 7238 of *Lecture Notes in Computer Science*, pages 228–242. Springer Berlin Heidelberg, 2012.

[4] Piers RJ Campbell and Ken Adamson. Methodologies for load forecasting. In *Intelligent Systems, 2006 3rd International IEEE Conference on*, pages 800–806. IEEE, 2006.

[5] Enrique Castillo, Diego Peteiro-Barral, Bertha Guijarro Berdiñas, and Oscar Fontenla-Romero. Distributed one-class support vector machine. *International journal of neural systems*, 25(07):1550029, 2015.

[6] Bo-Juen Chen, Ming-Wei Chang, et al. Load forecasting using support vector machines: A study on eunite competition 2001. *Power Systems, IEEE Transactions on*, 19(4):1821–1830, 2004.

[8] `http://www.sintef.no/Projectweb/DeVID/Prosjektpartnere/`

[7] Yu Cheng, Zhengzhang Chen, Lu Liu, Jiang Wang, Ankit Agrawal, and Alok Choudhary. Feedback-driven multiclass active learning for data streams. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 1311–1320, New York, NY, USA, 2013. ACM.

[8] Jui-Sheng Chou and Anh-Duc Pham. Smart artificial firefly colony algorithm-based support vector regression for enhanced forecasting in civil engineering. *Computer-Aided Civil and Infrastructure Engineering*, 30(9):715–732, 2015.

[9] EugeneA. Feinberg and Dora Genethliou. Load forecasting. In JoeH. Chow, FelixF. Wu, and James Momoh, editors, *Applied Mathematics for Restructured Electric Power Systems*, Power Electronics and Power Systems, pages 269–285. Springer US, 2005.

[10] Aldo Goia, Caterina May, and Gianluca Fusai. Functional clustering and linear regression for peak load forecasting. *International Journal of Forecasting*, 26(4):700 – 711, 2010.

[11] Alicia Mateo González, Antonio Muñoz San Roque, and Javier García-González. Modeling and forecasting electricity prices with input/output hidden markov models. *Power Systems, IEEE Transactions on*, 20(1):13–24, 2005.

[12] VictoriaJ. Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[13] TimothyM. Hospedales, Shaogang Gong, and Tao Xiang. Finding rare classes: Adapting generative and discriminative models in active learning. In JoshuaZhexue Huang, Longbing Cao, and Jaideep Srivastava, editors, *Advances in Knowledge Discovery and Data Mining*, volume 6635 of *Lecture Notes in Computer Science*, pages 296–308. Springer Berlin Heidelberg, 2011.

[14] Hao Huang, Qinming He, Kevin Chiew, Feng Qian, and Lianhang Ma. Clover: a faster prior-free approach to rare-category detection. *Knowledge and Information Systems*, 35(3):713–736, 2013.

[15] Slobodan A Ilić, Srđan M Vukmirović, Aleksandar M Erdeljan, and Filip J Kulić. Hybrid artificial neural network system for short-term load forecasting. *Thermal Science*, 16(suppl. 1):215–224, 2012.

[16] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.

[17] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[18] S. Lakshmivarahan. *Learning Algorithms Theory and Applications*. Springer-Verlag, 1981.

[19] Yuan-cheng LI, Ting-jian FANG, and Er-keng YU. Study of support vector machines for short-term load forecasting [j]. *Proceedings of the Csee*, 6:010, 2003.

[20] Petroula Louka, Georges Galanis, Nils Siebert, Georges Kariniotakis, Petros Katsafados, I Pytharoulis, and G Kallos. Improvements in wind speed forecasts for wind power prediction purposes using kalman filtering. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(12):2348–2362, 2008.

[21] Pengcheng Luo, Giovanni Sansavini, and Enrico Zio. Reliability analysis of smart grid communication network by simulation. *Integrated Computer-Aided Engineering*, 22(2):119–133, 2015.

[22] Vladimiro Miranda and Claudio Monteiro. Fuzzy inference in spatial load forecasting. In *Power Engineering Society Winter Meeting, 2000. IEEE*, volume 2, pages 1063–1068. IEEE, 2000.

[23] Mohamed Mohandes. Support vector machines for short-term electrical load forecasting. *International Journal of Energy Research*, 26(4):335–345, 2002.

[24] A.-H. Mohsenian-Rad and A. Leon-Garcia. Optimal residential load control with price prediction in real-time electricity pricing environments. *Smart Grid, IEEE Transactions on*, 1(2):120–133, 2010.

[25] K. Najim and A. S. Poznyak. *Learning Automata: Theory and Applications*. Pergamon Press, Oxford, 1994.

[26] K. S. Narendra and M. A. L. Thathachar. *Learning Automata: An Introduction*. Prentice-Hall, New Jersey, 1989.

[27] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis. Learning automata: Theory, paradigms, and applications. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 32(6):706–709, December 2002.

[28] M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis. Efficient fast learning automata. *Information Sciences*, 157:121–133, 2003.

[29] Angel Pardo, Vicente Meneu, and Enric Valor. Temperature and seasonality influences on spanish electricity load. *Energy Economics*, 24(1):55 – 70, 2002.

[30] TM Peng, NF Hubele, and GG Karady. Advancement in the application of neural networks for short-term load forecasting. *Power Systems, IEEE Transactions on*, 7(1):250–257, 1992.

[31] Tiago Pinto, Zita Vale, Tiago M Sousa, Isabel Praça, Gabriel Santos, and Hugo Morais. Adaptive learning in agents behaviour: A framework for electricity markets simulation. *Integrated Computer-Aided Engineering*, 21(4):399–415, 2014.

[32] A. S. Poznyak and K. Najim. *Learning Automata and Stochastic Optimization*. Springer-Verlag, Berlin, 1997.

[33] Hanne Saele, I Graabak, and Petter Stoa. Odel-standard format for exchange of metering and settlement information. In *Electricity Distribution, 2001. Part 1: Contributions. CIRED. 16th International Conference and Exhibition on (IEE Conf. Publ No. 482)*, volume 6, pages 5–pp. IET, 2001.

[34] Kjell Sand, Jan Foosnas, Dag Eirik Nordgard, Vidar Kristoffersen, Tarjei Benum Solvang, and Dagfinn Wage. Experiences from norwegian smart grid pilot projects. In *Electricity Distribution (CIRED 2013), 22nd International Conference and Exhibition on*, pages 1–4. IET, 2013.

[35] Carlos N Silla Jr and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

[36] L Suganthi and Anand A Samuel. Energy models for demand forecastingâĂŤa review. *Renewable and Sustainable Energy Reviews*, 16(2):1223–1240, 2012.

[37] James W Taylor. An evaluation of methods for very short-term load forecasting using minute-by-minute british data. *International Journal of Forecasting*, 24(4):645–658, 2008.

[38] M. A. L. Thathachar and P. S. Sastry. Varieties of learning automata: An overview. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 32(6):711–722, December 2002.

[39] M. A. L. Thathachar and P. S. Sastry. *Networks of Learning Automata: Techniques for Online Stochastic Optimization*. Kluwer Academic, Boston, 2003.

[40] M. L. Tsetlin. *Automaton Theory and the Modeling of Biological Systems*. Academic Press, New York, 1973.

[41] Selen Uguroglu. *Robust Learning with Highly Skewed Category Distributions*. PhD thesis, Carnegie Mellon University, 2013.

[42] Fernando VilariÃśo, Panagiota Spyridonos, Jordi VitriÃă, and Petia Radeva. Experiments with svm and stratified sampling with an imbalanced problem: Detection of intestinal contractions. In Sameer Singh, Maneesha Singh, Chid Apte, and Petra Perner, editors, *Pattern Recognition and Image Analysis*, volume 3687 of *Lecture Notes in Computer Science*, pages 783–791. Springer Berlin Heidelberg, 2005.

[43] Jianjun Wang, Li Li, Dongxiao Niu, and Zhongfu Tan. An annual load forecasting model based on support vector regression with differential evolution algorithm. *Applied Energy*, 94(0):65 – 70, 2012.

[44] Rafal Weron. *Modeling and forecasting electricity loads and prices: A statistical approach*, volume 403. Wiley. com, 2007.

[45] Zhongzhe Xiao, Emmanuel Dellandrea, Weibei Dou, and Liming Chen. Hierarchical classification of emotional speech. *IEEE Transactions on Multimedia*, 2007.

[46] Rong Yan, Yan Liu, Rong Jin, and Alex Hauptmann. On predicting rare classes with svm ensembles in scene classification. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 3, pages III–21–4 vol.3, 2003.

[47] Hong-Tzer Yang and Chao-Ming Huang. A new short-term load forecasting approach using self-organizing fuzzy armax models. *Power Systems, IEEE Transactions on*, 13(1):217–225, 1998.

[48] Guoqiang Zhang, B Eddy Patuwo, and Michael Y Hu. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62, 1998.

[49] Hans-Georg Zimmermann, Christoph Tietz, and Ralph Grothmann. Forecasting with recurrent neural networks: 12 tricks. In *Neural Networks: Tricks of the Trade*, pages 687–707. Springer, 2012.